A New Quality of Service Metric for Hard/Soft Real-Time Applications

Shaoxiong Hua and Gang Qu
Electrical and Computer Engineering Department and Institute of Advanced Computer Study
University of Maryland, College Park, MD 20742
{shua, gangqu}@glue.umd.edu

Abstract

Real-time applications often have mixed hard and soft deadlines, can be preempted subject to the cost of context switching or the restart of computation, and have various data dependency. The simple but widely used task completion ratio, as the Quality of Service (QoS) metric, does not capture these characteristics and can not reflect user perceived QoS well. In this paper, we propose a new quantitative QoS metric, which is based on task completion ratio but differentiates hard and soft deadlines and models data dependency as well. Basically, it assigns different weights to hard and soft deadline tasks, penalizes late soft task completion, and measures the tasks affected by any dropped tasks. We apply popular online schedulers, such as EDF (earliest deadline first), FCFS (first come first serve), and LETF (least execution time first), on a set of simulated MPEG movies at the frame level and for each application compare the new QoS measurement, traditional completion ratio with the "real" completion ratio which considers the number of correctly decoded frames and has been mapped to the user perceived QoS well. Experimental results show that our proposed QoS metric can reflect real life QoS much better than the traditional one.

1. Introduction

With the increasing popularity of real-time multimedia and wireless communication applications, quality of service (QoS) attracts a lot of attention in a number of research and development communities, in particular the network routing and multimedia delivery. Providing the required QoS guarantees are vital for design of the embedded systems that carry out such applications. The most popular way to specify time-related QoS requirements, such as synchronization and latency, is deadline. **Hard deadlines** are the deadline constraints that must be satisfied in order to provide QoS guarantees. However, as the application-driven system design keeps on pushing for high performance, light weight, low energy consumption, better portability, and so on, it becomes tough to meet these more and more system resource demanding QoS requirements. For example, one would like to view high-resolution movies one after another on a DVD player, but it cannot be done without recharging the battery. Consequently, many real-time applications have **soft deadlines**. Failure to meet the soft deadlines will degrade the QoS within an acceptable range. For instance, many MPEG video applications such as video conferences require reliable communication and consistently high throughput, while being able to tolerate reasonable amount of packet error, jitter, or unsynchronization. Missing some frame's (soft) processing deadline will also be acceptable. Soft deadlines can also be found in many other applications such as web browsing and file transfer.

In the other hand **data dependency** exists in many applications. The simple but widely used task completion ratio, as the Quality of Service, assumes that the tasks are independent and equally important. It cannot capture the hard/soft deadline and data dependency and does not reflect the user perceived QoS well.

These observations lead us to a new measure for QoS. Any task completion before its deadline should contribute to the QoS; any task completion after its soft deadline may also contribute to the QoS, but subject to a penalty for missing the deadline. Furthermore, the task drops will affect some other tasks based on data dependency. Therefore, we define QoS as a weighted sum of the percentage of completed tasks, the penalty for completing tasks after their soft deadlines and the penalty for task drops.

The paper is structured as follows. In Section 2 we briefly survey the existing work on QoS modeling. Section 3 proposes a novel quantitative measurement for QoS based on task completion ratio with consideration of penalties for task drops and soft deadline misses. Section 4 presents and analyzes the experiment results. The paper is concluded and some future envisioned research issues are presented in Section 5.

0-7695-1916-4/03 $17.00 © 2003 IEEE

2. Related work

Various QoS requirements, such as bounded delay, minimal throughput, guaranteed synchronization or resolution, task completion ratio, were first addressed in the network and real-time operating systems (RTOS) communities. Lawrence discusses the metrics based on the QoS attributes of timeliness, precision, and accuracy that can be used for system specification, instrumentation, and evaluation [6]. Altmann and Varaiya define QoS as a combination of the basic quality metrics for the network layer: delay, jitter, bandwidth, and reliability [1]. The most formally sound and practically relevant QoS model in the networking community is proposed by Cruz [4]. The model is based on the demand curves and service curves. The main conceptual result in RTOS literature is presented by Rajkumar et al. in [8]. They introduce an analytical approach for satisfying multiple QoS dimensions under a given set of resource constraints. They show that the problem is NP-hard and develop an approximation polynomial algorithm for the problem by transforming it into a mixed integer programming problem [9]. Comprehensive survey of QoS research in these two areas is given in [2]. Recently, Ng et al. report some research results on the QoS of MPEG video as perceived by Human beings. They derive a new metric QoS-Human for measuring the QoS of MPEG video of three common types of contents [7].

3. QoS model

We consider that a single processor system serves real-time applications, each application consists of a sequence of tasks, and each task is characterized by its arrival time *a*, deadline *d* (which can be either hard or soft), and execution time *e*. The lifetime of a task with hard deadline is the period *[a,d]* between its arrival time and deadline. The length of its lifetime, *d-a*, is normally referred as latency. We will represent a task by <*a,d,e,h/s*>. A task is completed if it receives valid service time in the amount of its required execution time, before the deadline for hard deadline task.

- A task has a **hard** deadline if it must be completed before the deadline otherwise the system will not get the reward for serving the task and the application.
- A task has a **soft** deadline if the system can still benefit even if the deadline is missed, subjected to a deadline-miss penalty.
- A task is **non-preemptive** means that once the task gets the CPU, it will occupy the CPU until its deadline or completion, whichever comes earlier.
- A task is **preemptive** means that the task may lose control of the CPU during its execution, but when it gets the CPU back, it can resume the interrupted execution.
- A task is **semi-preemptive** if it can be preempted but must be restarted instead of resuming the remaining workload due to the high cost of context switch. That is, any incomplete computation becomes invalid and will be discarded. For example, multimedia embedded systems may not have sufficient memory to store all the intermediate results and would restart the computation should preemption occurs.

The completion ratio, which is the percentage of how much tasks have been completed by the system, does not give an accurate measure for the QoS in that 1) it does not distinguish hard deadline tasks and soft deadline tasks on which the system may get different amount of reward; 2) it does not distinguish tasks completed before their soft deadlines and those that are completed but miss the deadlines; and 3) it does not reflect data dependency among tasks since all deadline misses are treated in the same way. Based on these observations, we define QoS as follows:

Suppose that a scheduler S completes K_h hard-deadline tasks and K_s soft-deadline tasks out of a total of N tasks, the QoS provided by such scheduler is:

$$Q(S) = \frac{\alpha_s K_s + \alpha_h K_h}{N} - \frac{\beta}{N}\sum \frac{\delta_i}{d_i - a_i} - \frac{\gamma}{N}\sum 1_i \Delta_i \qquad (1)$$

where α_s and α_h are the weights for soft deadline tasks and hard deadline tasks; β is the penalty parameter or the tolerance factor for deadline missing; δ_i is the difference between the task's deadline and completion time when the (soft) deadline is missed; d_i - a_i is the life time of the task; γ is penalty parameter for task dropping; Δ_i is the number of tasks that will be affected if the i-th task is dropped(1_i=1); the first sum is taken over all the completed tasks that miss their soft deadlines; and the second sum is taken over all the dropped tasks regardless of their deadline type.

The QoS defined in (1) is a direct extension of completion ratio, in the case when there is no penalty of missing soft deadlines (β=0) or dropping tasks (γ=0) and hard deadline tasks are considered equally important as soft ones (α_s=α_h=1), which has been used for QoS measurement in many occasions. Soft deadlines and hard deadlines are treated differently by assigning them different weights α_s and α_h. Soft deadline missing is penalized by the relative amount that the deadline has been missed with the penalty factor β. Data dependency is captured in the last term by reducing QoS in the amount of tasks depending on the dropped tasks with a penalty factor γ.

Table 1. Statistics of the MPEG streams we generated in the simulation. (μ is the mean and σ is the standard deviation.)

Movie	Numbers of frames	I-to-I	I-to-P	I-frame size		P-frame size		B-frame size	
				μ	σ	μ	σ	μ	σ
Wizard of OZ	41,700	15	3	15.18	13.61	4.82	0.64	3.91	0.27
Star Wars	174,960	12	3	8.68	5.51	3.93	0.58	2.81	0.52
Science of the Lambs	39,792	12	3	6.53	2.86	2.59	0.86	1.98	0.70
Goldfinger	40,104	12	3	9.77	6.60	4.57	0.51	3.26	0.38

4. Experimental results

In this section, we report the setup of our simulation on MPEG movies and comparison of different QoS measurements such as our proposed new QoS metric, the traditional completion ratio and the "real" completion ratio which considers the number of correctly decoded frames and has been mapped to the user perceived QoS well. Our results indicate that the new QoS metric can reflect the user perceived QoS much better than the traditional completion ratio.

4.1. Simulation of MPEG streams

We have applied popular online schedulers, such as EDF, FCFS, and LETF (least execution time first), on MPEG video streams decoding at the frame level. Standard MPEG encoders generate three types of compressed frames: I frames (intra-pictures), P frames (predicted pictures) and B frames (bi-directional predicted pictures) [10]. In general, encoders use a fixed GOP (Group of Pictures) pattern when compressing a video sequence. A typical GOP in display order and decoding order is shown as in Figure 1.

0 1 2 3 4 5 6 7 8 9 10 11 12
$I_0\ P_1\ B_2\ B_3\ P_4\ B_5\ B_6\ P_7\ B_8\ B_9\ I_{10}\ B_{11}\ B_{12}$ decoding order
$I_0\ B_2\ B_3\ P_1\ B_5\ B_6\ P_4\ B_8\ B_9\ P_7\ B_{11}\ B_{12}\ I_{10}$ display order

Figure 1. A typical GOP pattern(I-to-I=12, I-to-P=3)

On average, I frames are the largest in size (since they are self-contained), followed by P frames and B frames. Krunz and Tripathi present a comprehensive model for MPEG video streams [5]. The model captures the bit-rate variations at multiple time scales. Statistically, the generated MPEG streams fit the empirical video and are sufficiently accurate in predicting the queuing performance for real video streams.

From the parameters on four movies given in [5], we simulate their frame information that is reported in Table 1. (The frame size of I-frames has a relatively large standard deviation because it is modeled by the sum of two random components).

Based on the frame size and type, we generate the normalized execution time for each frame using a linear model of MPEG decoding [3]. In the simulation we assume that the inter-arrival time of frames are independent with exponential distribution. Its mean is approximately equal to the reciprocal of frame display rate (in terms of frame per second (fps)) to generate a balanced loaded system. We simulate underloaded and overloaded systems by varying the fps requirement. The deadline for decoding each frame is set corresponding to the arrival time and the frame display rate. We use several standard display rates in our simulation: 15, 24, 30, 45 and 60 fps. The deadline type is assigned to each individual frame based on the dependency of different frames. I frame is the most important, because the correct processing of all the P frame and B frame in the same GOP depends on the completion of the corresponding I frame. P frame is also important because it is required by the following P and B frames in the same GOP. We assign I and P frames hard deadlines rather than giving them soft deadlines. We also assign soft deadlines to B frames to create tasks with mixed type of deadlines.

Each GOP can be viewed as one "application" independent of others as the correct decoding of all the frames in one GOP depends on the leading I frame. Each "application" consists of a set of tasks (frame decoding) and the drop of hard deadline I and P frames will cause the incorrect decoding of the remaining frames in this "application". To better model the data dependency among "tasks", we assign different values Δ_I and $\Delta_{P,i}$, which are corresponding to the number of frames that will not be decoded correctly because of a dropped frame, to frames with hard deadlines. For example if I-to-I, the number of frames between two consecutive I frames (see Figure 1), is 11, then we assign $\Delta_I = 11$; $\Delta_{P,i}$ are assigned 10, 7, and 4 for the three P frames in the GOP pattern based on Figure 1; and $\Delta_B = 0$ because there is no frame depends on B frame. As a result, I frames have higher priority than P and B frames, P frames have higher priority than B frames. This exactly matches the MPEG decoding mechanism. In sum, we use the following QoS, based on formula (1) with consideration of MPEG application's characteristics, in our simulation:

$$Q_{MPEG}(s) = \frac{K_s + K_h}{N} - \frac{\beta}{N}\sum_{i=1}^{K_s}\frac{\delta_i}{T_d} - \frac{\gamma}{N}\left(m_I\Delta_I + \sum_{i=1}^{n_P} m_{P,i}\Delta_{P,i}\right) \quad (2)$$

$Where\quad T_d$ – *the reciprocal of frame display rate;*

$\Delta_I, \Delta_{p,i}$ – *the number of tasks that will be affected if the I frame or P frame is dropped;*

$m_I, m_{P,i}$ – *the number of dropping I, P frames;*

n_P – *the number of P frames in a GOP pattern;*

$K_s, K_h, \beta, \gamma, \delta_i, N$ *are same as in* (1).

4.2. Simulation results

We have implemented popular online scheduling algorithms, such as EDF, FCFS and LETF, and applied them to the simulated MPEG movies. For each movie, we simulate underloaded, balanced, and overloaded systems by changing the frame rate from 15 fps, to 24, 30, 45, and 60 fps. And for each case, we consider the case of non-preemptive, preemptive, and semi-preemptive. Now we detail the simulation results.

For underloaded system with a frame rate of 15fps and/or 24fps, the deadlines are relatively loose and we observe that almost all the algorithms achieve the maximal QoS in the amount of 1 without task drop and deadline missing. In this case, our proposed QoS metric is same as the traditional completion ratio. However, when the computation load increases, the system becomes balanced and overloaded eventually.

Figure 2-4 shows the relationship of traditional completion ratio, our proposed new QoS metric and "real" completion ratio, which considers the actual number of correctly decoded frames and has been mapped to the user perceived QoS well under different online scheduling policies (EDF and LETF) on different movies in the case of non-preemptive, preemptive and semi-preemptive for overeloaded system. In the simulation we assume that the frames arrive in the decoder order, so EDF and FCFS have the same results. From these figures we can see that the traditional completion ratio doesn't reflect the "real" completion

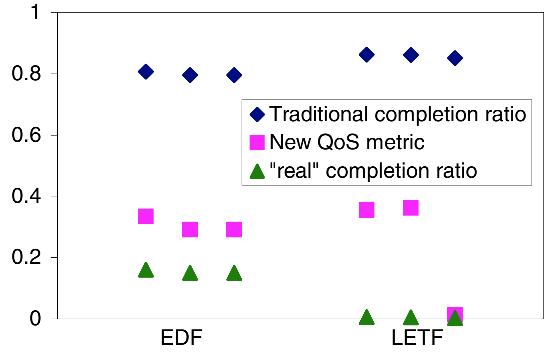

Figure 2. Comparison of three different QoS measurements under EDF and LETF on movie "Silence of the Lambs" in the frame rate of 45 fps in the case of, from left to right, non-preemptive, preemptive, and semi-preemptive.

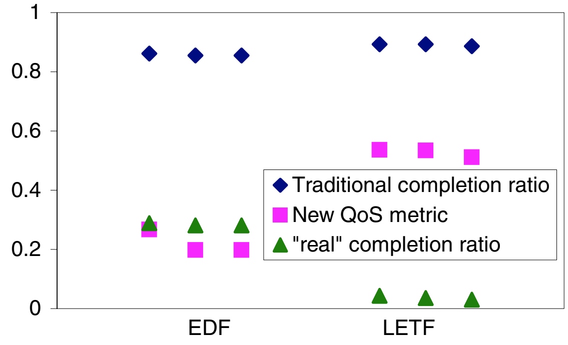

Figure 3. Comparison of three different QoS measurements under EDF and LETF on movie "Wizard of OZ" in the frame rate of 30 fps in the case of, from left to right, non-preemptive, preemptive, and semi-preemptive.

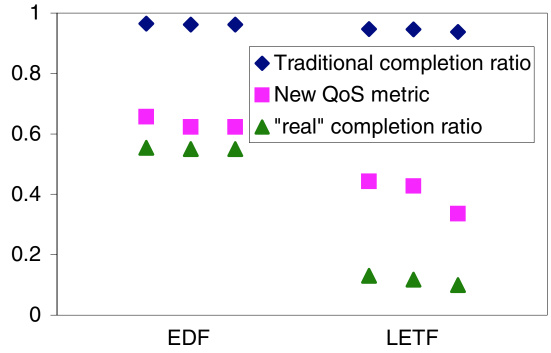

Figure 4: Comparison of three different QoS measurements under EDF and LETF on movie "Goldfinger" in the frame rate of 30 fps in the case of, from left to right, non-preemptive, preemptive, and semi-preemptive.

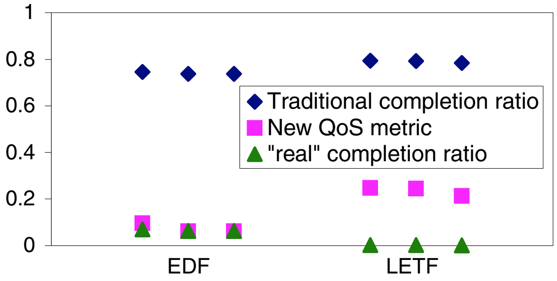

Figure 5. Comparison of three different QoS measurements under EDF and LETF on movie "Star Wars" in the frame rate of 45 fps in the case of, from left to right, non-preemptive, preemptive, and semi-preemptive.

ratio well. For example, in the Figure 2, although the traditional completion ratio is high, around 80%, the "real" completion ratio is less than 20% under EDF and only around 4% under LETF. Compared with traditional completion ratio, our proposed new QoS metric is much closer to the "real" completion ratio and reflects the user perceived QoS well. So it is necessary to develop low overhead online scheduler to maximize this new QoS metric in order to eventually improve the user perceived QoS without using extra hardware.

From these figures we also can observe that the "real" completion ratio under EDF is consistently better than that under LETF. The reason is that, in general, the execution time of B frame is shorter than that of I or P frame, therefore, for the LETF policy it prefers to select B frame which actually is the least important task. But the traditional completion ratio under LETF may be better than that under EDF, so the conclusion is that it is crucial to finish important tasks as many as possible, not the raw counter of task completions. For new QoS metric there is not clear relationship between EDF and LETF because it is affected by several factors such as the number of completed soft and hard tasks, the soft deadline missing, the task drop and the selection of penalty parameter β and γ.

Finally we can see that the QoS achieved in the case of non-preemptive is mostly higher than that achieved in the preemptive case and much higher than that achieved in the semi-preemptive case.

5. Conclusions

With the increasing popularity of real-time multimedia and wireless communication applications, quality of service (QoS) attracts a lot of attention. In this paper we present a new metric on how to measure the QoS provided by an embedded system to real-time applications with mixed hard and soft deadlines. It captures the mixed hard and soft deadline nature of such application, considers the innegligible preemption cost, and models data dependency. We apply popular online schedulers, such as EDF, FCFS and LETF, on a set of simulated MPEG movies at the frame level and find that for each application compared with the traditional completion ratio, our new QoS metric is much closer to the "real" completion ratio which considers the number of correctly decoded frames. And this indicates the new QoS metric can reflect user perceived QoS much better than the traditional completion ratio.

Further work is required to develop some new online schedulers with low runtime overhead to maximize the proposed new QoS in order to achieve better user perceived QoS without using extra hardware. We also anticipate designing systems that provide the same QoS guarantees with less system resources such as CPU, power, and memory.

6. References

[1] J. Altmann and P. Varaiya, "INDEX project: user support for buying QoS with regard to user's preferences", *Sixth International Workshop on Quality of Service (IWQoS'98)*, 1998, pp. 101-104.

[2] C. Aurrecoechea, A.T. Campbell, and L. Hauw, "A survey of QoS architectures", *Multimedia Systems*, Vol.6, No.3, May 1998, pp. 138-51.

[3] A. C. Bavier, A. B. Montz, and L. L. Peterson, "Predicting MPEG Execution Times", *ACM SIGMETRICS 98*, June 1998, pp. 131-140.

[4] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks", *IEEE Journal on Selected Areas in Communications*, Vol.13, No.6, August 1995, pp.1048-1056.

[5] M. Krunz and S.K. Tripathi, "On the characterization of VBR MPEG streams", *SIGMETRICS 97*, 1997, pp. 192-202.

[6] T. F. Lawrence, "The quality of service model and high assurance", *Proceedings High-Assurance Engineering Workshop*, 1997, pp. 38-39.

[7] J. K. Ng, K. Leung et al., "A scheme on measuring MPEG video QoS with human perspective", *Proceedings of the 8th International Conference on RTCSA,* 2002, pp. 233-241.

[8] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A resource allocation model for QoS management", *Proceedings. IEEE Real-Time Systems Symposium*, 1997, pp. 298-307.

[9] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. "Practical Solutions for QoS-based Resource Allocation Problems", *Proceedings. The 19th IEEE Real-Time Systems Symposium,* 1998, pp. 296-306.

[10] L. Teixeira and M. Martins, "Video compression: The MPEG Standards", *Proceedings ECMAST'96,* 1996, pp. 615-634.

0-7695-1916-4/03 $17.00 © 2003 IEEE